NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

CX1107 DS&A: Transform and Conquer **2021/22 Semester 1**

Lab 5: Algebraic Expression Representations

School of Computer Science and Engineering *Nanyang Technological University*

Q1 Write a C program to convert an infix expression to a postfix expression. The input and the output of the function are character strings. The input expression contains only four possible operators: +, -, * and /. Operands can be any alphanumeric. Each operand is represented by a character symbol. The parentheses are allowed in the input expression. You may assume that the expression is always valid. The function prototype is given below:

```
void in2Post(char* infix, char* postfix)
```

Some test sample cases:

1. Infix: (A+B)+(C-D)
   Postfix: AB+CD-+
2. Infix: a+b*c-d*(e/f)
   Postfix: abc*+def/*-
3. Infix: 9+(8-7)*(6/(5-4)+3)
   Postfix: 987-654-/3+*+

Q2 Write a C program to evaluate a postfix expression. You may assume that operands are single-digit numbers. The function prototype is given below:

```
double exePostfix(char* postfix)
```

Some test sample cases:

1. Input: 987-654-/3+*+
   Output: 18.00
2. Input: 25+89/9*7*+
   Output: 63.00
3. Input: 88/8*77+7+5*6*-
   Output: -622.00

Q3 Write a C program to convert an infix expression to a prefix expression.

```
void in2Pre(char* infix, char* prefix)
```

Some test sample cases:

1. Infix: (A+B)+(C-D)
   Prefix: ++AB-CD
2. Infix: a+b*c-d*(e/f)
   Prefix: -+a*bc*d/ef
3. Infix: 9+(8-7)*(6/(5-4)+3)
   Prefix: +9*-87+/6-543